

A Denotational Semantics Methodology (DSM) Approach for Business Processes Modeling

Roger Atsa Etoundi
Department of Computer Science
Faculty of Science
University of Yaoundé I

Marcel Fouda Ndjodo
Department of Computer Science
Faculty of Science
University of Yaoundé I

Ghislain Abessolo Alo'o
Department of Computer Science
Faculty of Science
University of Yaoundé I

ABSTRACT

In our previous work on business process, we have provided the foundation for formalization based on the environment concept. The environment is defined as a set of observers whose values change according to the actions of tasks. From their values, observers are grouped to form a set of states. A state is defined as the characteristic of an environment from which the satisfaction of the goal associated to the business process is checked. In this paper, based on Denotational Semantics, we will highlight the semantics core features used in our modeling approach. By applying the resulting models in daily work, enterprises will improve their productivity and quality of service in order to deal with the competitive pressure of the network economy.

Keywords

Business Process Modeling, Workflows, Denotational Semantics, Process Abstraction, Quality of Service.

1. INTRODUCTION

A business process refers typically to an enterprise process. Business process modeling (BPM) is one of the core methodologies which is developed to better represent the functional behavior of the information system dealing with the delivery of services to customers within an organization. BPM is not a formal approach in the line of the numerous models such as abstract system, abstract integration, system abstraction and simulation and concrete system. In fact, it rather belongs to the family of informal UML-like models, which seriously limits its theoretical potential and leaves the door open for new research [7].

The most important part of the ongoing research in business process management is focused on process modeling, since a process is one of the most important aspects in the delivery of goods and services within an enterprise. In business processes modeling, many techniques have widely been used [2, 15, 3, 17, 6, 4, 12]. Despite their popularity, there is no consensus on the modeling standard. The standard model of a business process and workflow should take into consideration perspectives such as process, organization, information and operation. Most research has been concentrated in the process perspective; putting aside other perspectives [13].

Naturally therefore, there are several important aspects of research that emerge in trying to develop formalism for business processes and more generally, for organization paradigms. Such research effort could practically help organizations to better master their structures and processes. In this regard, we have defined in [9, 11, 8] a framework that can be used as a starting point of the standardization of BPM. This approach, based on the concept of environment, allows capturing various abstractions of the perspectives defined above. The model we presented in our previous work does not describe the process semantics.

To refine our model, we will use the Denotational Semantics to express the semantics of concepts defined in [8]. Denotational Semantics has traditionally been described as the theory of what programs denote. In many cases, denotations are built with the help of functions in some mathematical universe. This presupposes that the ontological status of the domain, business process in our case, is firmly established. This ontology has been defined in [8, 10]. There has always been an alternative viewpoint in which Denotational Semantics is seen as a translation from one formalism into another in order to gain new insight into the object under consideration. This second position has become more and more popular over the last years due to the desire for syntactical descriptions of already existing mathematical objects. For this purpose, Denotational Semantics is applied in the business process area.

The rest of the paper is organized as follows: section 2 presents part of the Denotational Semantics concepts suitable for our modelling approach, section 3 presents our proposed model of a business process and the last section 4 concludes by highlighting some perspectives as future works.

1. DOMAIN OF MATHEMATICAL OBJECTS

In this section, we define parts of Denotational Semantics [1, 14] that are suitable in handling the modelling of business processes and workflows. We need some mathematical machinery. Most importantly, the domains used in mapping business processes and workflow concepts is presented. The domain of mathematical objects is defined in terms of partially ordered sets, least upper bounds, chains and continuity.

Definition 2.1 (Partially Ordered Set)

Let C be an arbitrary set. A partial order \subseteq on C is a subset of $C \times C$ which satisfies the following for all c_1, c_2 and c_3 in C :

1. $c \subseteq c$ (reflexivity),
2. if $c_1 \subseteq c_2$ and $c_2 \subseteq c_3$ then $c_1 \subseteq c_3$ (transitivity),
3. if $c_1 \subseteq c_2$ and $c_2 \subseteq c_1$ then $c_1=c_2$ (antisymmetry).

In this mathematical domain of objects, we are concerned not only with arbitrary sets with partial order, but also with sets of functions with their ordering. A partial ordering on a set of functions of type $C_1 \rightarrow C_2$ can be derived from the orderings on C_1 and C_2 .

Definition 2.2 (Sequence)

Let (C, \subseteq) be a partially ordered set, $\langle c_0, c_1, \dots \rangle$, also denoted by $\langle c_i \rangle_{i=0}^{\infty}$ is a sequence if and only if for all $i \in \mathbb{N}$:

$$c_i \subseteq c_{i+1}.$$

Definition 2.3 (CPO)

A complete partially ordered (CPO) set is a set C with a partial order \subseteq which satisfies the following requirements:

1. there is a least element, denoted by \perp , with respect to \subseteq , i.e $\forall c \in C: \perp \subseteq c$,
2. each sequence $\langle c_i \rangle_{i=0}^\infty$ in C has a least upper bound in C.

Definition 2.4 (Continuous Function)

Let C_1, \subseteq_1 , C_2, \subseteq_2 be two CPO's, the function

$f : C_1 \rightarrow C_2$ is continuous if and only if for each chain

$\langle c_i \rangle_{i=0}^\infty$ in C, the following holds:

$$f \bigcup_{i=0}^\infty c_i = \bigcup_{i=0}^\infty f c_i$$

Fact 1 (Fixed Point Theorem) let C be a CPO and $f : C \rightarrow C$,

if f is continuous, then the least fixed point μf exists and is

equal to $\bigcup_{i=0}^\infty f^i \perp$, where $f^0 \perp = \perp$

and $f^{i+1} \perp = f f^i \perp$.

Definition 2.5 (Least Fixed Point)

Let C, \subseteq be a CPO, $f : C \rightarrow C$ and let $x \in C$.

- x is a fixed point of f if $f(x)=x$
- x is a least fixed point of f if x is a fixed point of f and for each fixed point y of f , the relation $x \subseteq y$ holds.

Definition 2.6 (Partial Ordering Functions)

Let C_1, \subseteq_1 , C_2, \subseteq_2 be two partially ordered sets, $C_1 \rightarrow C_2$ the set of continuous functions, an ordering \subseteq on $C_1 \rightarrow C_2$ is defined as follows, where $f, g \in C_1 \rightarrow C_2$:

$$f \subseteq g \Leftrightarrow \forall c \in C_1 : f c \subseteq_2 g c.$$

Definition 2.7 (Least Upper Bound)

Let $C' \subseteq C$. $c \in C$ is called a least upper bound of C' if:

1. c is an upper bound of C' i.e $\forall x \in C', x \subseteq c$,
2. c is a minimal element of the set of upper bounds of C' i.e $\forall y \in C: ((\forall x \in C': x \subseteq y) \Rightarrow c \subseteq y)$.

The least upper bound of a partially ordered set C' will be denoted by $\bigcup C'$.

Definition 2.8 (Least upper Bound of a Sequence)

The least upper bound of a sequence $\langle c_0, c_1, \dots \rangle$ denoted by

$\bigcup_{i=0}^\infty c_i$ or $\bigcup \langle c_i \rangle_{i=0}^\infty$ is defined as follows:

$\bigcup \langle c_i \rangle_{i=0}^\infty = \bigcup c \mid c \in \langle c_i \rangle_{i=0}^\infty$ where c in

$\langle c_i \rangle_{i=0}^\infty$ means that c is an element of the sequence $\langle c_i \rangle_{i=0}^\infty$.

Fact 2 (CPO of functions) Let C_1, \subseteq_1 and C_2, \subseteq_2 be two CPO's, then $C_1 \rightarrow C_2, \subseteq$ is a CPO.

Fact 3 (Least upper bound of functions) Let C_1, \subseteq_1 and

C_2, \subseteq_2 be CPO's and let $\langle f_i \rangle_{i=0}^\infty$ be a chain of functions in

$(C_1 \rightarrow C_2)$, then the function $\lambda_{c_1} \bigcup_{i=0}^\infty \langle f_i c_1 \rangle$ is the least upper bound of this chain and therefore

$$\bigcup_{i=0}^\infty f_i c_1 = \bigcup_{i=0}^\infty f_i c_1 \text{ for all } c_1 \in C_1.$$

In the rest of this paper, μf denotes the least fixed point of f where f is a continuous function on a CPO.

3. DSM APPROACH

In this section, the concepts of business processes and workflows are given in terms of their mathematical models.

3.1 The Environment Description Model

In the modelling of business processes proposed by a great number of researchers, such as van der Aalst [15] and Jorg Becker and al. [2], one can notice that they have neglected the effect that an environment can produce in the achievement of a given business goal within an organization. For example, in developing countries, moving from one hospital to another, we will realize that the manner of carrying out surgeries differ even if the surgeons graduate from the same institute. This is due to the fact that hospitals are not equipped in the same manner. In the modelling of a business process, the environment within which it will be performed has to be taken into consideration.

Without any loss of generality, we consider that an environment is a set of different metrics whose value may change [10]. These metrics are primitive Boolean observers denoted by Observer. The associated value of each observer depends on the current state of the environment.

Formally, an environment E is defined as a couple $\langle \Theta, S, val \rangle$ where:

- Θ is a non empty set of observers;
- S is a non empty set of sates;
- $val : \Theta \rightarrow S \rightarrow Bool$ is a function which describes the behaviour of observers.

In the rest of this paper, we denote $val(o)(s)$ by $s(o)$ where s

denotes a state and o an observer, $s(o)$ is the value of the observer o in the state s . Given a state s , the set of observers whose value is true defines the characteristic of s and is represented by $s_c = \{o \in \Theta, s(o) = true\}$.

Given two states s_1 and s_2 of the set of states S of the environment E , the set of observers whose associated values are not the same is defined from the characteristics of the two states. This set is called gap between s_1 and s_2 and is denoted by $s_1 \ominus s_2 = s_1 - s_2 \cup s_2 - s_1$.

Given an environment E , the observers in Θ define the alphabet that permits to reason about events that occur on E . The language defined from this alphabet is denoted by the set of conditions or formulae C . A condition $c \in C$ is an assertion over observers and is defined as a first order formula. The basic elements of C are therefore all the observers of Θ . The elements of C are formed by the following:

$$\left\{ \begin{array}{l} \text{if } o \in \Theta, \text{ then } o \in C \\ \text{if } o \in \Theta, \text{ then } \neg o \in C \\ \text{if } o_1, o_2 \in C, \text{ then} \\ \quad o_1 \wedge o_2, o_1 \vee o_2, o_1 \Rightarrow o_2 \in C \end{array} \right.$$

A condition c can be decomposed into a set of observers $+c$ whose values are evaluated to true and a set of observers $-c$ that are evaluated to false. The two sets do not have any common element i.e. $+c \cap -c = \emptyset$.

Given a condition $c \in C$ and a state s , c is satisfied within the state s if the result of its evaluation is true, i.e. $s(c) = true$.

3.1.1 State of an Environment

A state is a snapshot of an environment within a time. From this snapshot facts are observed. Some of these facts or features of a state are true or false at this particular time. These facts are represented as some equivalent of predicate calculus formulae. We shall refer, somewhat loosely, to these facts and relations as attributes of a state. In a rigorous manner, let F be a set of formulae, and s be a state, then s is a subset of F i.e. $s \in F$.

In general, let S be a set of states, according to the definition of a state, (S, \subseteq) is a partial ordered set. In our work, we are not dealing with any kind of set of states, we are interested with S

having a least state, \perp_S , known as initial state of a business process or workflow from which the execution can be started. This initial state is therefore contained in all states of S i.e for all $s \in S$, $\perp_S \subseteq s$. In the meantime, S is required to have a

least upper bound \bigcup_S known as a state where the goal of the business process is satisfied. As such, the set of states of a business process is mapped to a CPO concept.

3.2 Task Description Model

A task is an atomic activity that cannot be split into smaller activities. The performance or execution of a task transforms the state of the environment into another state. A task is therefore an action within a state of an environment. Before a task can be executed, the state of the environment should satisfy a specific condition called pre condition, and when this execution is completed another condition, called post condition is satisfied. A task is formally defined by a triple $\langle nt, pre, post \rangle$ where nt denotes the name of the task, pre its pre condition, and $post$ its post condition.

The action of a task within an environment is to transform its current state into a new one. When $\langle t, pre, post \rangle$ is a task, s a given state where the precondition pre is satisfied i.e $s(pre) = true$, the action of t in the state s is the new state $t(s)$ which satisfies the post condition $post$ i.e $t(s)(post) = true$. In general, the action of a task t within the state s is characterised by the observers of s whose value has been modified.

Definition 3.1 (Task action)

Let $E = \langle \Theta, S, val \rangle$ be an environment, s a given state and t a task whose pre condition is satisfied in s , then the action of t in s denoted by t_s and is specified by $t_s = \{o \in \Theta, s(o) \neq t_s(o)\}$.

When there will be no ambiguity, a task will be represented by its name t and $pre(t)$ respectively $post(t)$ will denote respectively its pre and post condition. Based on the post condition of a task t , and the state s where $s(post(t)) = true$, we conjecture that $t_s = +post(t) \cup -post(t)$.

Definition 3.2 (Conflicting Tasks)

The action of tasks within an environment can be conflicted as many tasks can modify the same observers at the same time. To this end, t_1 and t_2 are conflicting tasks in the state s , and we denote it by $overlap(t_1, t_2, s)$, if and only if:

$$\left\{ \begin{array}{l} s(pre\ t_1) = s(pre\ t_2) = true \\ +post\ t_1 \cap -post\ t_2 \neq \emptyset \\ +post\ t_2 \cap -post\ t_1 \neq \emptyset \end{array} \right.$$

Definition 3.3 (Shift)

Let SoT be a none empty set of tasks and s a given state, a shift denoted by Shf is a couple $Shf = \langle s, SoT \rangle$ composed with the state s and the set of non conflicting tasks SoT within s .

Formally, let $Shf = \langle s, SoT \rangle$ be a shift, the following properties are satisfied:

$$\left\{ \begin{array}{l} S \circ T \neq \emptyset \quad 1 ; \\ \forall t \in S \circ T, s \text{ pre } t = \text{true} \quad 2 ; \\ \forall t, t' \in S \circ T, t \neq t' \Rightarrow \\ \quad \text{overlap } t, t', s = \text{false} \quad 3 . \end{array} \right.$$

Let $Sht = \langle s, SoT \rangle$ be a shift, the simultaneous actions of SoT in s , denoted by $ts(s)$, is captured by the set of observers whose values are modified within s , that is:

$$S \circ T \ s = \bigcup \left\{ \begin{array}{l} o \in \Theta : \\ o \in -\text{post } t_i \end{array} \right\}, t_i \in S \circ T$$

Definition 3.4 (Chain)

A chain is an execution path of tasks, according to their actions in states and their triggering conditions is denoted by $P = \prod_{i=1}^n Sht_i$, and is specified as a finite sequence of shifts where n represents the length of the sequence.

Let P be a path of length $n > 1$, and

$$sh_k = \langle s_k, st_k \rangle, sh_{k+1} = \langle s_{k+1}, st_{k+1} \rangle \text{ notes}$$

respectively the shift in the range k and $k+1$, the state s_{k+1} is the resulting state after the execution of the set of tasks st_k i.e

$s_{k+1} = st_k \ s_k$. When there will be no ambiguity, the shift of the range k of the path P will be denoted by $P \ k$.

$$\text{Let } Sht_k = \langle S_k, S \circ T_k \rangle \text{ and } Sht_{k+1} = \langle S_{k+1}, S \circ T_{k+1} \rangle$$

be two shifts where $Sht_k = S \circ T_k \ s_k$, the difference between the states s_k and s_{k+1} is denoted by $\overline{s_k + s_{k+1}}$ and is defined as follows: $\overline{s_k + s_{k+1}} = S \circ T_k \ s_k$.

Lemma 3.1

Let p be an execution path and $t \in S \circ T \ p \ k$ with

$k \leq \text{length } p$ then there will always exist m such that $m > k$ and $S(p(m))(post(t)) = \text{false}$.

Lemma 3.2

Let p be an execution path

then $S \circ T \ p \ \text{length } p = \emptyset$.

Definition 3.5 (State ordering)

Let P be a path of length $n > 1$, and $Sht_k = \langle S_k, S \circ T_k \rangle$ and

$Sht_{k+1} = \langle S_{k+1}, S \circ T_{k+1} \rangle$ be two consecutive shifts in P with $k < n$ then $S_k \subseteq S_{k+1}$ specifies the fact that the set of observers modified in S_k after the actions of SoT are contained in the set of observers of S_{k+1} with the same values.

Lemma 3.3

Let P be an execution path, S the set of states of P , then

S, \subseteq is CPO where the least upper bound state in the last state of P and the least state is the first state of P .

At the level of this modelling, we have to ensure that the execution of a task t will stop at a certain time. In order to do so, the set of observers that should be modified by t must contain partially or totally in the observers forming its pre condition

$-\text{pre } t \cup -\text{pre } t \cap -\text{post } t \cup +\text{post } t \neq \emptyset$ From the definition of the execution path of tasks, we specify the relation within the set T of tasks based on the set S of states. This relation is denoted by \trianglelefteq .

Definition 3.6 (Ordering of Tasks)

Let T be a set of tasks, and $t1$ and $t2$ be two tasks of T , we write $t1 \trianglelefteq t2$ if and only if for all chain CH such that if n_{t1} and n_{t2} denote respectively the maximum range of $t1$ and $t2$ in CH , then $n_{t1} \leq n_{t2}$.

This relation has the following properties:

1. reflexivity: $t \trianglerighteq t$ this simply means that the task t belongs to the chain CH ;
2. antisymmetric: if $t_1 \trianglerighteq t_2$ and $t_2 \trianglerighteq t_1$ in the chain D then $t_1 = t_2$. By convention, there will always exist a path from each task to itself;
3. transitivity: obviously if in the chain CH , $t_1 \trianglerighteq t_2$ and $t_2 \trianglerighteq t_3$ then $t_1 \trianglerighteq t_3$.

Lemma 3.4

The set of tasks T associated with the relation previously defined \trianglerighteq , i.e. T, \trianglerighteq , forms a complete partial ordered set.

3.2.1 Palette

Definition 3.7

Let E be an environment, and S be a set of different states that E may reach according to the actions of tasks T , then a palette P is a couple $\langle E, S \rightarrow S \rangle$. In the rest of this paper, the set of functions $S \rightarrow S$ will be denoted by T , the set of tasks of the palette. When there will be no ambiguity, $P(E)$ and $P(E)$ will denote the environment and the set of tasks of the palette P respectively.

The actions of the set of tasks T of the palette P in the environment E are to change at least once the value of each observer of Θ in E . To this end, the consecutive actions of a non empty set of tasks within an environment may not modify all the observers in this environment. The set of observers whose value are not changed during the execution of any given none empty set of tasks will be abstracted from all the possible states of the environment, i.e.

$$\forall o \in \Theta \left\{ \begin{array}{l} \exists t_1 \in T, o \in -post\ t_1 \quad or \\ \exists t_2 \in T, o \in +post\ t_2 \end{array} \right.$$

Given a palette P , according to the environment changes within organizations and the different executions of tasks that can take place, different ways in which tasks can be executed have to be captured. In the rest of the paper, we will be using SPP to specify the set of execution paths that can be obtained from a palette P .

Lemma 3.5

Let P be a palette, $s \in S\ P$ a given state of the environment

$E(P)$ of P , there will always exist a path $p \in SP_p$ such that

$s \in S\ p$, where $S\ p$ denotes the set of states of the path p .

Lemma 3.6

Let $P=\langle E, T \rangle$ be a palette, and $t \in T$, there will exist an execution path $ch \in SP_p$ where SP_p denotes the set of

possible execution paths of T , $ch\ n = \langle s_n, S \circ T_n \rangle$ such that $t \in S \circ T_n$.

3.3 Business Process Model

A business process is a collection of activities or tasks designed to produce a specific output for customers. It implies a strong emphasis on how work is done within an organization in order to deliver a particular service. A process is thus a specific order of work activities across time and space, with a beginning, an end, and clearly defined inputs and outputs. The output is the reason the organization does this work and is defined in terms of the benefits this process has for the organization as a whole.

Definition 3.8 (A service)

A service is the characteristic of a business process and is defined as a composition of a set of criteria that characterize what is delivered within an organization, where each criterion is represented by an observer.

The model of a business process is defined as a couple $\langle P, G \rangle$ where P is a palette and G the service to be achieved. According to the definition of the palette, the ordering of tasks is captured explicitly by their pre conditions and the states of the environment within which their execution is being carried out.

This approach reduces the number of patterns to be used in order to capture various ways tasks can be ordered. We strongly believe that this is the main difference between our approach and other BPM theory papers presented in the literature. In these works, the Workflow Management Coalition [6] has identified four basic

control structures for workflows: *OR-SPLIT*, *OR-Join*, *AND-Split*, and *AND-Join*. more control structures have been identified by Van der Aalst in [16]. We argue that these control flow structures can be formally captured by the first order formula. In our future work, we will be presenting these concepts using the denotation defined above.

Lemma 3.7

There will always exist a state S_{lub} such that when it is reached, other states cannot be reached. This state is called a least upper bound state of the associated business process.

Lemma 3.8

There will always exist a state S_{ini} from which the execution of the business process starts. This state is called a least state of the associated business process.

For each service associated to a given business process, a set of qualities of service is defined to deal with the daily work and the competitive pressure of the network economy.

3.4 QoS Model

The quality of service denoted by QoS represents the performances of the service which determine the level of satisfaction projected for the recipients of the services. The level of satisfaction is defined as a set of properties, criteria, characteristics and performances of the services delivered to the customers. Several works are made in this field, each one defining a specific set of criteria specified in order to measure the QoS. In the literature, there is no consensus yet on the definition of the set of common criteria to evaluate the quality of service delivered in the organizations. The evaluation criteria are defined according to the objectives and specificities of each company. In this work, we define an abstract model which gives the semantics of the quality of service.

Definition 3.9 Let Cr be a set of criteria considered in the evaluation of the quality of service, Val the set of values that can be assigned to these criteria, and f a map defined by $f: C \rightarrow Val$, the QoS is defined by (C, Val, f) .

Given two QoS q_1 and q_2 such that $q_1=(Cr_1, Val_1, f_1)$ and $q_2=(Cr_2, Val_2, f_2)$, we say that q_1 and q_2 are compatible and we note $q_1 \Delta q_2$ if and only if $C_1=C_2$ and $Val_1=Val_2$. When q_1 and q_2 are compatible, we will say that q_1 is better to q_2 and we note $q_1 \sqsubseteq q_2$ if and only if $\forall c \in C_1\ f_1(c) \leq f_2(c)$. In the rest of the paper, we use (ϕ, \sqsubseteq) to denote the partial ordered set of compatible qualities of services.

Definition 3.10 (Well Defined Business Process)

Let, $BP=\langle P, G \rangle$ be a business process, BP is well defined if and only if all the observers that form its goal (service) are contained in the set of observers of the environment E i.e.

$$-G \cup +G \subseteq \Theta\ E .$$

Definition 3.11 (Well Formed Business Process)

Let $BP=\langle P, G \rangle$ be a business process, BP is said to be well

formed if and only if each execution chain SCH reaches the least upper bound state s_{lub} which satisfies the service G i.e.

$$\left\{ \begin{array}{l} \forall ch \in SCH, n_{ch} \in N, s_{lub} \in S \\ n_{ch} = length \ ch \\ s_{lub} \\ s_{lub} \ G = true \end{array} \right.$$

More formally, let SCH be the non empty set of different chains that can be obtained from a business process BP , and

$CH \in SCH$ with the length n_{CH} such that the n_{CH}^{th} state

s_{lub} of CH satisfies G i.e. $s_{lub} = true$.

Definition 3.12 (Deadlock- and Livelock-Free)

Let BP be a business process, BP is deadlock- and livelock-free if and only if it guarantees that every execution chain reaches its least upper bound state satisfying the goal of the business process BP .

Theorem 3.1

Let BP denote a business process such that BP is well defined and well formed, then BP is deadlock-free and livelock-free.

Proof: By the definitions of well formedness and well definedness of a business process which states that the least upper bound of the state of a business process is reached and that this least upper bound state satisfied the goal of the business, we can then conclude that the described business process model is deadlock and live lock free.

All the execution paths of a business process start from the same state denoted by s_{ini} . It can be easily being shown that the set of states S_{BP} associated with the ordering relation \subseteq as defined previously is CPO.

3.5 Human Actor Model

There are many types of agents participating in the processing of tasks within an enterprise for the achievement of customers' needs. The enterprise system dealing with the processing of tasks is a hybrid system including hardware components with embedded software, the human actors interacting with the hardware and the organization. An organization is an arrangement of human actors purposefully organized to carry out a certain mission, which, in its turn, adds a dimension to the quality of service. The hardware components have been designed to play specific roles and functions in the process chain, and can hardly be moved among different roles in the enterprise as it is done for human actors. In this work, we are not dealing with hardware but with human actors who can significantly influence the quality of service according to their skills and associated experiences. We model the skill of a human actor by (Sk, Tks, mch) where Sk is the set of competences, Tks the set of tasks and mch a map that gives for each competence $cp \in Sk$ the set of tasks $mch(cp) \in Tks$ that

can be processed based on cp with $mch(cp) \neq \emptyset$. When there will be no ambiguity the structure (Sk, Tks, mch) will be represented by Sk . Based on the organization put in place, the set of tasks assigned to a human actor are kept in a diary.

A diary is described by the set of tasks and the set of time intervals within which there are processed. We require that the set of time intervals in the agenda be defined such that it does not allow the overlapping of time intervals.

Let $Pds = (TI, \subseteq, \cap, \Delta)$ be a set of time intervals such that (TI, \subseteq) is a partial ordered set with ∂ the smallest time interval, \cap and Δ be two maps defined as follows $\cap : TI \times TI \rightarrow TI$ and $\Delta : TI \times TI \rightarrow Boolean$, $t1$ and $t2$ be two time intervals of TI , $p1$ and $p2$ overlapped if and only if there exists a time interval $p3$ such that:

$$p1 \cap p2 = t3 \Rightarrow \begin{cases} p3 \Delta p1 \wedge p3 \Delta p2 \\ p3 \subseteq p1 \wedge p3 \subseteq p2 \end{cases}$$

where \cap and Δ define respectively the intersection and the overlapping relationship. When there will be no ambiguity, the set of time intervals will be represented by Pds . Based on the concepts of tasks and time interval, the diary concept is modeled by $\langle Tks, Pds, g \rangle$ where Tks is the set of tasks, Pds the set of associated time intervals, and g a map defined by $g: Tks \rightarrow Pds$ such that $\forall t1, t2 \in Tks, t1 \neq t2 \Rightarrow \neg(g(t1) \Delta g(t2))$.

Definition 3.13 (Human Actor)

A human actor is defined by $\langle Sk, Ex, f, Dy, Id \rangle$ where Sk is its set of skills, Ex the set of associated experiences, Id its identification, Dy its associated diary, and f a map which defines for each skill $sk \in Sk$ its associated experience $f(sk) \in Ex$.

3.6 Enterprise Model

An enterprise is a structure dealing with the service delivery of customers based on a certain quality of service. This structure is organized in terms of business processes that are carried out, employees in charge of the processing of the associated tasks, and the resulting workflows.

Definition 3.14 (Enterprise) An enterprise Org is modeled by $(Io, BPs, Emps, WFs)$ where Io is its identification, BPs is the set of its business processes that can be run, $Emps$ its set of employees who participated in the processing of tasks, WFs its set of workflows.

Definition 3.15 (Workflow) A workflow is defined by $(Ts, Es, Ps, h, Q)^+$ where Ts is the set of none conflicting tasks, Es the set of employees dealing with the processing of Ts within the time intervals Ps to obtain the quality of service Q , and h is the map $Ts \rightarrow Ps$ which defines for each task t , its time interval $h(t)$ within which it is processed.

Based on the human actors working in a given enterprise and their availability and the services required by customers, employees involved in different workflows associated to a business process will not necessary be the same. To this end, according to their skills, the quality of service delivered may be different. The criteria for the evaluation of the quality of service will then some time be associated with minimum values when tasks will be processed by staff with minimum experience. More-over these values will be maximal when staff with maximum experience have been involved in the processing of tasks. The set of quality of service associated to a given business process will therefore

have two specific qualities of service Q_{\min} and Q_{\max} which have the following properties.

Lemma 3.9 Let $Q_{\min} = (C, \text{Val}, f_{q_{\min}})$, and $Q_{\max} = (C, \text{Val}, f_{q_{\max}})$, be minimal and the maximal quality of service of a business process (P, \emptyset) then $\forall p = (C, V, fp) \in \emptyset, c \in C, f_{q_{\min}}(c) \leq fp(c)$, and $\forall q = (C, V, fq) \in \emptyset, c \in C, fq(c) \leq f_{q_{\max}}(c)$.

4. CONCLUSION

The main technical content of this paper is to present an abstract formal model for workflows, quality of service and business processes using Denotational Semantics for enterprise modeling. The business process model presented in this paper takes into consideration the environment or the context of its execution and the quality of service that can be delivered by the enterprise. The main concepts in defining this model is the environment and the quality of service. We have captured the salient features of an environment which are observers and states. From these features we have defined in an incremental manner the model of a business process within an enterprise.

According to models defined using other approaches, our model does not explicitly define the dependency among tasks. We argue that from the definition of tasks by their pre and post conditions, the relation between tasks can be obtained. This will be part of our future works. Based on the notion of chain, we have built the execution path of a business process. Each of these paths is required to reach the least upper bound state of the environment which satisfies the associated goal of a business process. This goal is defined as a set of criteria that are assigned for the evaluation of the quality of service. From the concept of chain, we have defined the well formedness and well definedness properties of a business process. We use these definitions to show that a business process is deadlock and livelock free. We complete our modeling by giving an abstract representation of an enterprise based on the business processes and human actors who deal with the processing of associated tasks in order to deliver a given quality of service. We do believe that the defined models can serve as a guideline for business and workflows modelling within various organizations. We would like in future works, to carry out investigation based on many case studies, what features should be added to the model according to practical needs for the development of supporting tools.

5. ACKNOWLEDGMENT

The authors wish to thank the Ministry of the Public Service and Administrative Reform for its immense financial contribution. This work is being used to ameliorate the quality of service rendered to the State's Personnel.

6. REFERENCES

- [1] Jesper Andersen, Ebbe Elsborg, Fritz Hengle, Jakob Grue Simonsen and Christian Stefansen, "Compositional specification of commercial contracts", *Springer verlag*, 2006
- [2] Jorg Becker, Michael rosemann, and Chrisoph von Uthmann. "Business Process Management", Guideline of Business Process Modeling. *Springer-Verlag Berlin Heidelberg*, 2000
- [3] F. Casati, S. Ceri, B. Pernici, and G. Pozzi. "Conceptual Modeling of Workflows". *Springer Verlap*, December 1995

- [4] Amit P. sheth, Will van der Aalst, Ismailcem B. Arpinar. "Processes driving the networked economy". *IEEE Concurrency*, July-September 1999
- [5] C.R.Ramakrishnan H. Davulcu, M. Kifer and I.V. Ramakrishnan. "Logic based modeling and analysis of workflows". *ACM Symposium on Principles of Database systems*, pages 25–33, june 1998
- [6] D. Hollinsworth. "The workflow reference model". Technical Report TC00-1003, Workflow Management Coalition, <http://www.aiai.ed.ac.uk/WfMC/>, December 1994
- [7] Tomasz Janowski and Ojo Adegboyege. "Formalising Feasibility and correctness of distributed business processes". *Lecture Notes in Computer Science*, number 2465, pages 432–443, 2002
- [8] Daniel Krob. "Modelling of complex software systems: a reasoned overview". *International conference on formal methods for networked and distributed systems*, 2006
- [9] Atsa Etoundi Roger Marthe Monessa, Marcel Fouda Ndjodo and Erick Zobo. Feature-Oriented workflow modelling based on enterprise human resource planning. *Business process management journal*, 12(5):608–621, 2006
- [10] Atsa E. Roger and Marcel Fouda. "An Abstract Model For Workflows and Business Processes", *CARI 2002*, pages 239–247
- [11] Atsa Etoundi Roger. "A Domain Engineering Approach for Multi Perspective Business Process and Workflows Modeling", *PhD thesis, University of Yaounde I*, December 2004
- [12] Atsa Etoundi Roger and Marel Fouda Ndjodo. "A Generic Abstract Model for Business Processes and Workflows Management", *Bieter Gerald and Kirste Thomas, editors, 4th International Workshop on Mobile Computing*, pages 62–72. IRB Verlag, Stuttgart Germany, 2003
- [13] Oumaima Saidani and Selmin Nurcan. "Role-based approach modelling flexible business process", *Business Process Modeling, Development and Support*, 2006
- [14] R.D. Tennent. The denotational semantics of programming languages. Communication of the ACM, 1976
- [15] W. van der Aalst and al. "Business process Management", Techniques for modeling Workflows and Their Support of Reuse, pages 1–15. *Springer Verlag Berlin Heidelberg*, 2000
- [16] W. M. P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. "Workflow patterns", Technical Report wp 47, BETA Research Institute, 2000
- [17] W.M.P. van der Aalst, K.M. van Hee, and G.J. Houben. "Modelling and analysing workflow using a Petri-net based approach", *Proceedings of the second workshop on computer supported cooperative work*, pages 31–50, 1995